Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Argumentation-Based Recommendations: Fantastic Explanations and How to Find Them*

Antonio Rago, Oana Cocarascu, Francesca Toni
Department of Computing, Imperial College London, UK
{a.rago15, oc511, ft}@imperial.ac.uk

Abstract

A significant problem of recommender systems is their inability to explain recommendations, resulting in turn in ineffective feedback from users and the inability to adapt to users' preferences. We propose a hybrid method for calculating predicted ratings, built upon an item/aspect-based graph with users' partially given ratings, that can be naturally used to provide explanations for recommendations, extracted from user-tailored Tripolar Argumentation Frameworks (TFs). We show that our method can be understood as a gradual semantics for TFs, exhibiting a desirable, albeit weak, property of balance. We also show experimentally that our method is competitive in generating correct predictions, compared with state-of-the-art methods, and illustrate how users can interact with the generated explanations to improve quality of recommendations.

1 Introduction

Recommender systems [Resnick and Varian, 1997] aim to help users discover items that may be of interest. The most widely used types of methods for recommender systems are 'collaborative filtering' (looking at similar users and their preferences for determining recommendation to users), 'content-based filtering' (operating on information about users and their tastes) and 'hybrid' methods (combining the two). These methods use the vast amount of available data in a way that human users alone will never be able to. However, these systems suffer from scalability, data sparsity, 'cold-start' problems and lack of explanations for recommendations. The latter is an issue because if the reasoning behind recommendations is not explained to users then the feedback they provide may be ineffective in helping the system adapt to users' preferences, which in turn may cause users' unwillingness to follow the recommendations in the future.

In this paper we give a hybrid method for calculating predicted ratings for items in a recommender system and show how this method can be used to address the problem of recommender systems of providing explainable recommendations with which users can naturally interact. We show experimentally that our method significantly outperforms various state-of-the-art algorithms, namely Singular Value Decomposition [Billsus and Pazzani, 1998; Vozalis and Margaritis, 2006], Non-negative Matrix Factorization [Luo *et al.*, 2014], Slope One [Lemire and Maclachlan, 2007], Coclustering [George and Merugu, 2005], as well as KNN [Altman, 1992] and KNN with z-score normalization of each user.

Our method is built upon an item/aspect-based graph with users' partially given ratings. We show that these graphs can be mapped onto user-tailored Tripolar Argumentation Frameworks (TFs) which may be seen as instances of 'tripolar frameworks' as defined in [Gabbay, 2016] and of 'generalised argumentation frameworks' as defined in [Baroni *et al.*, 2017], and extend abstract [Dung, 1995] and bipolar [Cayrol and Lagasquie-Schiex, 2005] frameworks by including a 'neutralising' relation in addition to the standard 'attack' and 'support'. The mapping of item/aspect-based graphs onto user-tailored TFs is determined by predicted ratings for the user, calculated using our method. These predicted ratings can be understood as a gradual semantics for the TF, exhibiting a desirable, albeit weak, property of balance.

We illustrate how user-tailored TFs can then be used to give explanations that can help elicit users' feedback leading to positive effects on the quality of future recommendations.

The paper is organised as follows. In Section 2 we provide background on recommender systems, explanation in recommender systems, argumentation as understood in AI, as well as existing work on using argumentation to provide explanations for recommender systems. In Section 3 we define item/aspect-based graphs and provide our method for calculating predicted ratings. In Section 4 we experiment with a movie dataset, drawn from Netflix and imdbapi. In Section 5 we map item/aspect-based graphs onto user-tailored TFs and illustrate explanation and feedback incorporation. In Section 6 we conclude, in particular pointing to future work.

2 Background

The main methods used in recommender systems are 'latent factor models' and 'neighbourhood models' between items or users. Latent factor models, based on matrix factorization, describe the items as vectors of factors inferred from data. Neighbourhood models have been used to support various collaborative filtering algorithms for recommender systems. These models include non-negative matrix factoriza-

*Adapted from movie 'Fantastic beasts and where to find them'.

tion models [Luo *et al.*, 2014], Singular Value Decomposition [Billsus and Pazzani, 1998; Vozalis and Margaritis, 2006], Slope One techniques [Lemire and Maclachlan, 2007], and Co-clustering, a simultaneous clustering of users and items [George and Merugu, 2005]. In addition, collaborative filtering and content-based filtering can be combined to give hybrid models [Burke, 2002; 2003]. The Netflix Prize competition[1] has shown that matrix factorization models are superior to nearest-neighbour models, such as KNN, as, indeed, many of the best performing algorithms in the competition were based on matrix factorization [Koren *et al.*, 2009; Töscher *et al.*, 2009]. Whilst these models are scalable and effective, they are not easily explainable, as the way they represent factors makes them non-interpretable. We show that our proposed model is competitive with respect to the state-of-the-art, and how it lends itself to providing explanations.

[Tintarev and Masthoff, 2007] give an overview of explanations in recommender systems and identify four desirable features of recommender systems: *transparency*, by explaining how systems work and showing how they predict ratings; *scrutability*, by allowing feedback based on these explanations; *trust*, by correcting the systems based on user feedback; and *effectiveness*, by increasing the systems' accuracy with regards to users' preferences. None of the systems surveyed in [Tintarev and Masthoff, 2007] *fulfilled* all four of these aims. Of those which aimed to improve scrutability, [Billsus and Pazzani, 1998; Czarkowski, 2006] both use template responses based on factors affecting the recommendation. Our method identifies these factors and the reasons why they play a role by generating user-tailored argumentation frameworks.

Abstract argumentation frameworks (AFs) are pairs consisting of a set of arguments and a binary relation between arguments, representing attacks [Dung, 1995]. Formally, an AF is any $\langle AR, attacks\rangle$ where $attacks \subseteq AR \times AR$. Bipolar argumentation frameworks (BFs) extend AFs by considering two separate binary relations between arguments: attack and support [Cayrol and Lagasquie-Schiex, 2005]. Formally, a BF is any $\langle AR, attacks, supports\rangle$ where $\langle AR, attacks\rangle$ is an AF and $supports \subseteq AR \times AR$. Various other types of argumentation frameworks have been proposed in the literature, including tripolar frameworks, as in [Gabbay, 2016], and generalised argumentation frameworks, as in [Baroni *et al.*, 2017], both allowing for additional dialectical relations (in addition to attack and support). The argumentation frameworks we use in the paper for explanation can be seen as a special instance of these latter types of frameworks.

Several argument-based recommender systems have been proposed in the literature. For example, some [Chesñevar *et al.*, 2009; Briguez *et al.*, 2014; Teze *et al.*, 2015] use Defeasible logic programming (DeLP) [García and Simari, 2004] to enhance recommendation technologies with argument-based analysis. DeLP uses defeasible reasoning dialectically, can handle incomplete and contradictory information, and uses a comparison criterion to solve conflicting situations between arguments. [Chesñevar *et al.*, 2009] models user preferences as facts, strict rules and defeasible rules. Along with background information, user preferences can be used in a DeLP program to make recommendations which are modelled as arguments in favour of or against a particular decision. [Teze *et al.*, 2015] enhances the argument-based recommender system of [Chesñevar *et al.*, 2009] to allow for an argument comparison criterion on user's preferences to be encoded by means of conditional expressions. The movie recommender system of [Briguez *et al.*, 2014] relies on a set of predefined postulates describing the conditions in which a movie should be recommended to a given user and which can be translated into DeLP rules. Examples of postulates are "A user may like a movie if the actor of the movie is one of the user's favorite ones" or "A user may like a movie if the movie is liked by a group of similar users". Explanations are extracted from the dialectical tree supporting a recommendation. Our argumentation-based explanations are generated automatically from data without any need for knowledge to be manually incorporated.

[1] https://www.netflixprize.com/

3 The Aspect-Item Recommender System

We consider recommender systems where *items* (e.g. movies) are associated with *aspects* (e.g. comedy), which in turn have *types* (e.g. genre), and *users* may have provided *ratings* on some of the items and/or aspects. We refer to the frameworks underlying these recommender systems as *aspect-item*:

Definition 1 An *Aspect-Item framework* (A-I) is a 6-tuple $\langle \mathcal{I}, \mathcal{A}, \mathcal{T}, \mathcal{L}, \mathcal{U}, \mathcal{R}\rangle$ such that:

- $\mathcal{I}$ is a finite, non-empty set of *items*;
- $\mathcal{A}$ is a finite, non-empty set of *aspects* and $\mathcal{T}$ is a finite, non-empty set of *types*, where each aspect in $\mathcal{A}$ has a unique type in $\mathcal{T}$; for any $t \in \mathcal{T}$, we use $\mathcal{A}_t$ to denote $\{a \in \mathcal{A} |$ the type of a is $t\}$;
- the sets $\mathcal{I}$ and $\mathcal{A}$ are pairwise disjoint; we use $\mathcal{X}$ to denote $\mathcal{I} \cup \mathcal{A}$, and refer to it as the set of *item-aspects*;
- $\mathcal{L} \subseteq (\mathcal{I} \times \mathcal{A})$ is a symmetrical binary relation;
- $\mathcal{U}$ is a finite, non-empty set of *users*;
- $\mathcal{R} : \mathcal{U} \times \mathcal{X} \to [-1, 1]$ is a partial function of *ratings*.

We assume that ratings, when defined, are real numbers in the [-1,1] interval. Other types of ratings can be translated into this format, for example a rating $x \in \{1, 2, 3, 4, 5\}$ can be translated into a rating $y \in [-1, 1]$ using $y = ((x - 1)/2) - 1$.

The $\mathcal{I}$, $\mathcal{A}$, $\mathcal{T}$ and $\mathcal{L}$ components of an A-I may be visualised as a graph, as illustrated in Figure 1 for the movie domain.

In the remainder of the paper we assume as given a generic A-I $\mathcal{F} = \langle \mathcal{I}, \mathcal{A}, \mathcal{T}, \mathcal{L}, \mathcal{U}, \mathcal{R}\rangle$, unless otherwise specified.

Definition 2 The set of *linked item-aspects* of $x \in \mathcal{X}$ is $\mathcal{L}(x) = \{y \in \mathcal{X} | (y, x) \in \mathcal{L}\}$. We also use $\mathcal{L}_t(i)$ to denote $\{a \in \mathcal{L}(i) | a \in \mathcal{A}_t, i \in \mathcal{I}\}$.

For the example shown in Figure 1, the set $\mathcal{L}_{actor}$(Catch Me If You Can) comprises Leonardo DiCaprio and Tom Hanks.

The primary models for the goals of recommender systems [Aggarwal, 2016], formulated for A-Is, are: (i) Prediction - for a user $u \in \mathcal{U}$, $\forall i \in \mathcal{I}$ such that $\mathcal{R}(u, i)$ is undefined, compute a *predicted rating* $\mathcal{P}^u_{\mathcal{I}}(i)$; and (ii) Ranking - for a user $u \in \mathcal{U}$, compute a *ranking* on $\{i \in \mathcal{I} | \mathcal{R}(u, i)$ is undefined$\}$. We focus on prediction. Before giving our method for predicting ratings, we define users' *profiles*.

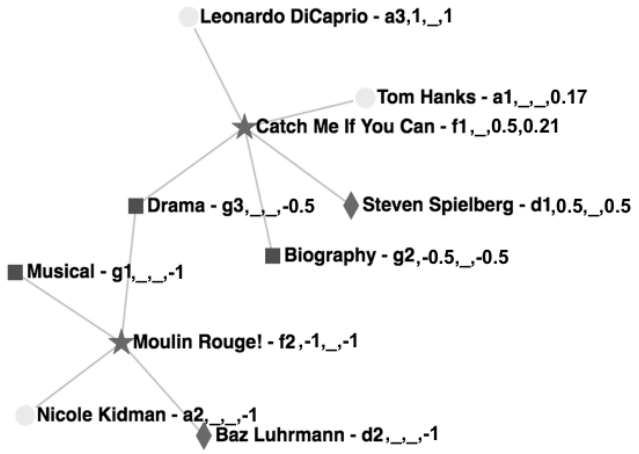

Figure 1: Example components of an A-I visualised as a graph, with items given by stars and types: **genres** (whose aspects are squares), **actors** (whose aspects are circles) and **directors** (whose aspects are diamonds). Each node's label is of the form (Name - x, $\mathcal{R}(u,x)$, $\mathcal{R}(v,x)$, $\mathcal{P}^u_{\mathcal{X}}(x)$), with $\mathcal{U} = \{u, v\}$ and _ standing for 'undefined'.

Definition 3 The *profile* π_u of user $u \in \mathcal{U}$ consists of:
- a *'collaborative filtering'* constant $\mu^u_{c.f.} \in [0,1]$;
- $\forall t \in \mathcal{T}$ a *'type importance'* constant $\mu^u_t \in [0,1]$;
- $\forall v \in \mathcal{U}$ such that $u \neq v$, a *'similarity'* constant $\omega_{u,v} \in [0,1]$.

Intuitively, $\mu^u_{c.f.}$ defines how much u wishes collaborative filtering to be taken into account, and a larger $\mu^u_{c.f.}$ will give other users' ratings more prevalence in the calculations of predicted ratings. Also, μ^u_t defines how important type t is to u and how much u wants aspects of type t to be taken into account, and larger values of μ^u_t will give these aspects, and the user's own ratings on items which are linked to them, a higher impact. Finally, $\omega_{u,v}$ defines how similar u and v are, and how much $v's$ ratings should impact the calculations.

Our method for calculating predicted ratings of items, based on users' profiles, makes use of the following notion of *weighted average rating*:

Definition 4 For any $u \in \mathcal{U}$ and any $i \in \mathcal{I}$, let $\Upsilon_u(i) = \{v \in \mathcal{U}\backslash\{u\} | \mathcal{R}(v,i) \text{ is defined}\}$ be the set of users other than u who have rated item i. Then, the *weighted average rating* $\rho^u : \mathcal{I} \to [-1,1]$ is obtained as follows, for $u \in \mathcal{U}$ and $i \in \mathcal{I}$:

if $\Upsilon_u(i) \neq \varnothing$ and $\sum_{v \in \Upsilon_u(i)} \omega_{u,v} > 0$

then $\rho^u(i) = \frac{\sum_{v \in \Upsilon_u(i)} \omega_{u,v} \mathcal{R}(v,i)}{|\Upsilon_u(i)|}$.

Thus, the weighted average rating of an item for a user is undefined when no other user or no other similar users have given any ratings for the item.

The predicted rating for an item is given in terms of the predicted rating for aspects, defined as follows.

Definition 5 For any user $u \in \mathcal{U}$ and aspect $a \in \mathcal{A}$, let $\Lambda^u(a) = \{i \in \mathcal{L}(a) | \mathcal{R}(u,i) \text{ is defined}\}$ be the set of linked items with ratings from u and let $\Lambda^{-u}(a) = \{i \in \mathcal{L}(a) | \rho^u(i) \text{ is defined}\} \backslash \Lambda^u(a)$ be the set of linked items with defined weighted average ratings but without ratings from u. Then, the *predicted aspect rating* $\mathcal{P}^u_{\mathcal{A}} : \mathcal{A} \to [-1,1]$ for a is obtained as follows, for $u \in \mathcal{U}$ and $a \in \mathcal{A}$:

if $\mathcal{R}(u,a)$ is defined then $\mathcal{P}^u_{\mathcal{A}}(a) = \mathcal{R}(u,a)$; else

if $\Lambda^u(a) = \Lambda^{-u}(a) = \varnothing$ then $\mathcal{P}^u_{\mathcal{A}}(a) = 0$; else

if $\Lambda^u(a) = \varnothing$ then

$$\mathcal{P}^u_{\mathcal{A}}(a) = \mu_{c.f.} \frac{\sum_{i \in \Lambda^{-u}(a)} \rho^u(i)}{|\Lambda^{-u}(a)|} / [1 + \mu_{c.f.}]; \text{ else}$$

if $\Lambda^{-u}(a) = \varnothing$ then

$$\mathcal{P}^u_{\mathcal{A}}(a) = \frac{\sum_{i \in \Lambda^u(a)} \mathcal{R}(u,i)}{|\Lambda^u(a)|}; \text{ else}$$

$$\mathcal{P}^u_{\mathcal{A}}(a) = [\frac{\sum_{i \in \Lambda^u(a)} \mathcal{R}(u,i)}{|\Lambda^u(a)|} + \mu_{c.f.} \frac{\sum_{i \in \Lambda^{-u}(a)} \rho^u(i)}{|\Lambda^{-u}(a)|}] / [1 + \mu_{c.f.}]$$

Intuitively, the predicted aspect rating weights the average ratings on linked items from the user and from similar users based on $\mu_{c.f.}$, but is overridden by a rating on the aspect itself from the user. Aspects without ratings (from the user or similar users) have the neutral predicted aspect rating of zero.

We finally use the predicted aspect ratings to calculate the predicted item ratings, as follows.

Definition 6 For any $u \in \mathcal{U}$, the *predicted item rating* $\mathcal{P}^u_{\mathcal{I}} : \mathcal{I} \to [-1,1]$ is obtained as follows, for any $i \in \mathcal{I}$:

if $\mathcal{R}(u,i)$ is defined then $\mathcal{P}^u_{\mathcal{I}}(i) = \mathcal{R}(u,i)$; else

if $\rho^u(i)$ is defined and $\sum_{t \in \mathcal{T}} \mu_t = 0$ then $\mathcal{P}^u_{\mathcal{I}}(i) = \rho^u(i)$; else

if $\rho^u(i)$ is undefined and $\sum_{t \in \mathcal{T}} \mu_t > 0$ then

$$\mathcal{P}^u_{\mathcal{I}}(i) = \frac{\sum_{t \in \mathcal{T}} \mu_t [\sum_{a \in \mathcal{L}_t(i)} \mathcal{P}^u_{\mathcal{A}}(a)] / |\mathcal{L}_t(i)|}{\sum_{t \in \mathcal{T}} \mu_t}; \text{ else}$$

if $\rho^u(i)$ is defined and $\mu^u_{c.f.} + \sum_{t \in \mathcal{T}} \mu_t > 0$ then

$$\mathcal{P}^u_{\mathcal{I}}(i) = \frac{\mu^u_{c.f} \rho^u(i) + \sum_{t \in \mathcal{T}} \mu_t [\sum_{a \in \mathcal{L}_t(i)} \mathcal{P}^u_{\mathcal{A}}(a)] / |\mathcal{L}_t(i)|}{\mu^u_{c.f.} + \sum_{t \in \mathcal{T}} \mu_t}; \text{else}$$

$\mathcal{P}^u_{\mathcal{I}}(i) = 0$

The predicted item rating is again overridden by a rating from the user. This calculation weights the average ratings on the item from similar users with $\mu_{c.f.}$ against the predicted aspects ratings from each of the linked aspects using their corresponding μ_t. Thus, aspects with a positive, negative or neutral predicted ratings have positive, negative or neutralising, respectively, effects on items to which they are linked. Note that our method can be seen a form of hybrid recommender system as it combines collaborative filtering with content-based factors.

In the remainder of the paper, for simplicity we use $\mathcal{P}^u_{\mathcal{X}}(x)$ to refer to $\mathcal{P}^u_{\mathcal{I}}(x)$ or $\mathcal{P}^u_{\mathcal{A}}(x)$ depending on whether $x \in \mathcal{I}$ or $x \in \mathcal{A}$, respectively. We also refer to $\mathcal{P}^u_{\mathcal{X}}$ as the *predicted rating* of an item-aspect.

As an illustration, consider the A-I with $\mathcal{I}$, $\mathcal{A}$, $\mathcal{T}$ and $\mathcal{L}$ as in Figure 1, $\mathcal{U} = \{u, v\}$ and $\mathcal{R}$ such that: $\mathcal{R}(u, a_3) = 1$, $\mathcal{R}(u, d_1) = 0.5$, $\mathcal{R}(u, f_2) = -1$, $\mathcal{R}(u, g_2) = -0.5$ and $\mathcal{R}(v, f_1) = 0.5$. Assume that $\mu_{c.f.} = \mu_{actors} = \mu_{genres} = \mu_{directors} = 1$ and $\omega_{u,v} = 0.5$. Then, by Definitions 5 and

6, the predicted rating for the item-aspects that u has rated is equal to these ratings, e.g. $\mathcal{P}^u_{\mathcal{A}}(u, a_3) = \mathcal{R}(u, a_3) = 1$ (likewise for d_1, f_2 and g_2). For a_1, $\Lambda^u(a_1) = \varnothing$ and $\Lambda^{-u}(a_1) = \{f_1\}$ thus $\mathcal{P}^u_{\mathcal{A}}(a_1) = \mu^u_{c.f.} \times \omega_{u,v} \times \mathcal{R}(v, f_1)[1 + \mu^u_{c.f.}] = 1 \times 0.5 \times 0.5/[1 + 0.5] = 0.167$. For x any of a_2, d_2 and g_1, $\mathcal{P}^u_{\mathcal{A}}(u, x) = \mathcal{R}(u, f_2) = -1$ since $\Lambda^u(x) = \{f_2\}$ and $\Lambda^{-u}(x) = \varnothing$. For g_3, $\Lambda^u(g_3) \neq \varnothing$, $\Lambda^{-u}(g_3) \neq \varnothing$ and $\mathcal{P}^u_{\mathcal{A}}(g_3) = [\frac{-1}{1} + 1\frac{0.5*0.5}{1}]/[1 + 0.5] = -0.5$. Finally, for f_1:

$$\mu^u_{c.f}\rho^u(f_1) = \mu^u_{c.f.} \times \omega_{u,v} \times \mathcal{R}(v, f_1) = 1 \times 0.5 \times 0.5 = 0.25;$$

$$\mu^u_{actors}[\sum_{a \in \mathcal{L}_{actors}(f_1)} \mathcal{P}^u_{\mathcal{A}}(a)]/|\mathcal{L}_{actors}(f_1)| = 1 \times [0.167 + 1]/2 = 0.584;$$

$$\mu_{genres}[\sum_{a \in \mathcal{L}_{genres}(f_1)} \mathcal{P}^u_{\mathcal{A}}(a)]/|\mathcal{L}_{genres}(f_1)| = 1 \times [-0.5 - 0.5]/2 = -0.5;$$

$$\mu_{directors}[\sum_{a \in \mathcal{L}_{directors}(f_1)} \mathcal{P}^u_{\mathcal{A}}(a)]/|\mathcal{L}_{directors}(f_1)| = 1 \times [0.5]/1 = 0.5;$$

$$\mathcal{P}^u_{\mathcal{I}}(f_1) = \frac{0.25 + 0.584 - 0.5 + 0.5}{4} = 0.209.$$

These predicted ratings, alongside the given ratings, if any, are visualised in Figure 1.

4 Evaluation on a Movie Dataset

We evaluate experimentally the A-I recommender System on a movie dataset extracted from the Netflix dataset[2] and imdbapi[3]. The Netflix dataset consists of 17K movies with over 100 million datestamped (beween 31st Dec 1999 - 31st Dec 2005) 5-star ratings by 480K users. Our movie dataset contains information about 528 movies with 500 reviews each, giving a total of 260K ratings from 37K users; the movies are those in the Netflix dataset in which some 745 most popular/star actors have acted, as obtained from imdbapi.[4] In addition to these 745 actors, our dataset contains information about 389 directors and 20 movie genres, also collected using imdbapi. Finally, as in all illustrations until now, we focus on three types: genre, actors and directors. This is based on the fact that most users prefer certain genres, they follow specific actors, or are interested in the work of some directors. Formally, in our experiments, we consider various A-Is $\langle \mathcal{I}, \mathcal{A}, \mathcal{T}, \mathcal{L}, \mathcal{U}, \mathcal{R} \rangle$ where $|\mathcal{I}| = 528$, $|\mathcal{U}| = 37K$, $\mathcal{T} = \{\text{genre, actor, director}\}$ and the other components, as well as the association of types with aspects, are straightforwardly obtained from the original datasets.

In our experiments, we use the following constants for the profile of all users $u \in \mathcal{U}$: $\mu^u_{c.f.} = 0.3$, $\mu^u_{genre} = 0.3$, $\mu^u_{actor} = 0.5$, $\mu^u_{director} = 0.2$, and, to determine the 'similarity' constants between any two (different) users, we use the cosine distance between the users' preferences for all aspects of type genre. Formally, $\omega_{u,v} = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\| \cdot \|\boldsymbol{v}\|}$ where $\boldsymbol{u}$ and $\boldsymbol{v}$ are vectors representing user u's and user v's preferences, respectively, for each aspect a in $\mathcal{A}_{genre}$. In the experiments, for each u, we use this definition of $\omega_{u,v}$ only for v any of the most similar 20 users to u, and use $\omega_{u,v} = 0$ for all other v.

Model	Min #movies training set/ #movies 'cold-start'			
	10/5	20/5	20/7	20/10
Co-clustering	83.4%	84.1%	85.1%	86.7%
KNN	85.5%	85.7%	85.9%	86.6%
KNN with z score	85.5%	85.3%	86.4%	87.5%
NMF	83.7%	84.2%	85.3%	86.1%
Slope one	86.2%	86.0%	87.2%	88.2%
SVD	85.9%	86.3%	87.3%	87.8%
A-I model	**94.9%**	**94.0%**	**93.3%**	**93.4%**

Table 1: Experimental accuracy results on our movie dataset using various baseline algorithms, with training sets including different numbers of minimum movies seen by users and various numbers of movies to address the 'cold-start' problem.

Further, in our experiments we vary the number of users in the training set, considering all users who have rated at least 10 movies or, alternatively, 20 movies, and make predictions for users who have rated fewer than 10 or 20 movies, respectively (in other words these users are part of our test set). To address the 'cold-start' problem, our training sets also include 5, 7 or 10 movies for users who have rated fewer than 10 or 20 movies, with these (5, 7 or 10) movies not included in the test sets. The movies rated by users who overall rated fewer than 5, 7 or 10 movies all belong to the training sets.

Since ratings are highly subjective, users who might like the same movie could give different ratings, e.g. two users who both liked a movie could give 5 and 4 stars, respectively. Thus, in all of our experiments, we consider predicted ratings differing from an actual rating by 1 star to be suitable to cater for variations in subjective judgement.

We use various algorithms as baselines, implemented using the Suprise library [Hug, 2017]: Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), Slope One, KNN and KNN with z-score normalization of each user, and Co-clustering. Our approach, the A-I model, outperforms all other methods in terms of accuracy in all experiments (using the subset of the Netflix dataset we have considered), as reported in Table 1. Note that the average root-mean-square error across the four experiments for the A-I model, looking at actual vs. predicted ratings, was 0.58.

The experiments show that the A-I model can make competitive predictions. In the next section we show how, under this model, A-Is may be given an argumentative reading from which explanations can be extracted to elicit and integrate users' feedback for improved predictions over time.

5 Argumentative Explanations

In abstract [Dung, 1995] and bipolar [Cayrol and Lagasquie-Schiex, 2005] argumentation, any information which may be in dialectical relationships of disagreement (attack) or, in the

[2] https://www.netflixprize.com/

[3] http://www.theimdbapi.org

[4] We use popularity of actors as a filter to obtain movies that, having been seen by more users, may be divisive, while still including less popular movies from actors' early careers. The resulting dataset is still of ample size to explore our method's potential.

bipolar case, agreement (support) with other information may be considered to be an argument, and arguments (according to this loose interpretation of the term) typically have a negative or positive impact on the (gradual) acceptability of arguments they attack or support, respectively. In this spirit, item-aspects in A-Is may be seen as arguments: if a user (or another similar user) rates an item highly/lowly then this item can be seen as an argument for/against, respectively, the aspects connected with the item and, similarly, if a user rates an aspect highly/lowly then this aspect can be seen as an argument for/against, respectively, the items connected with the aspect. Moreover, if an A-I is viewed from an argumentative perspective, a user's (or similar user's) opinion (rating) on an aspect/item may impact the estimation of the user's opinion (rating) of items/aspects connected with that aspect/item in the absence of actual ratings. This argumentative reading of A-Is facilitates the extraction of explanations for predictions that users can fruitfully interact with to provide feedback.

In order to capture A-Is as argumentation frameworks, however, a novel dialectical *neutralising* relationship is needed, in addition to the standard relationships of attack and support as in bipolar argumentation frameworks, to represent item-aspects which have neither a positive nor a negative effect on other arguments but rather *neutralise* them, by moving their strength towards the middle point:

Definition 7 A *Tripolar Argumentation Framework* (TF) is a quadruple $\langle \mathcal{X}, \mathcal{L}^-, \mathcal{L}^+, \mathcal{L}^0 \rangle$ where $\mathcal{X}$ is a set of *arguments*, and $\mathcal{L}^-$, $\mathcal{L}^+$, $\mathcal{L}^0$ are binary relations over $\mathcal{X}$. For $x, y \in \mathcal{X}$, we say that x *attacks* y if $(x, y) \in \mathcal{L}^-$, x *supports* y if $(x, y) \in \mathcal{L}^+$, and x *neutralises* y if $(x, y) \in \mathcal{L}^0$. With $\times$ as any of $-$, $+$ or 0, for any $x \in \mathcal{X}$, we will use $\mathcal{L}^\times(x)$ to denote $\{y \in \mathcal{X} | (y, x) \in \mathcal{L}^\times\}$.

Note that our TFs may be seen as instances of 'tripolar frameworks' as defined in [Gabbay, 2016] and of 'generalised argumentation frameworks' as defined in [Baroni *et al.*, 2017]. Whereas these works envisage the use of relations other than attack and support we commit (in our concrete instance) to the additional relation 'neutralise'.

Straightforwardly, any TF $\langle \mathcal{X}, \mathcal{L}^-, \mathcal{L}^+, \mathcal{L}^0 \rangle$ with $\mathcal{L}^0 = \varnothing$ is a bipolar argumentation framework and if $\mathcal{L}^+ = \mathcal{L}^0 = \varnothing$ then the TF is an abstract argumentation framework. As in the case of abstract and bipolar argumentation frameworks, a TF may also be equipped with some gradual evaluation method σ which calculates the strength of any argument over a given interval based on the strength of the arguments in dialectical relationships with the argument, as in [Gabbay, 2016].

We map an A-I onto a user-tailored TF, as follows. First we *direct* the A-I's links in $\mathcal{L}$, based on the existence of the user's and other (similar) users' ratings for item-aspects:

Definition 8 The *directed A-I* for $u \in \mathcal{U}$ is $\mathcal{F}^u = \langle \mathcal{I}, \mathcal{A}, \mathcal{T}, \mathcal{L}^u, \mathcal{U}, \mathcal{R} \rangle$, where $\mathcal{L}^u = \{(i, a) \in \mathcal{L} | \mathcal{R}(u, i)$ is defined or $\exists v \in \mathcal{U}$ such that $\mathcal{R}(v, i)$ is defined and $\omega_{u,v} \neq 0\} \cup \{(a, i) \in \mathcal{L} | \mathcal{R}(u, i)$ is undefined$\}$. For $x \in \mathcal{X}$, we refer to $\mathcal{L}^u(x) = \{y \in \mathcal{X} | (y, x) \in \mathcal{L}^u\}$ as the set of item-aspects *affecting* x. Also, for $i \in \mathcal{I}$ we use $\mathcal{L}^u_t(i)$ to denote the set $\{a \in \mathcal{L}^u(i) | a \in \mathcal{A}_t\}$.

For the remainder of the paper we will assume as given a generic directed A-I $\mathcal{F}^u = \langle \mathcal{I}, \mathcal{A}, \mathcal{T}, \mathcal{L}^u, \mathcal{U}, \mathcal{R} \rangle$ for $u \in \mathcal{U}$, unless otherwise specified. A TF can then be obtained from $\mathcal{F}^u$ by determining the polarity of pairs in $\mathcal{L}^u$, as follows:

Definition 9 For any $i \in \mathcal{I}$, let $r^u(i)$ be $\mathcal{R}(u, i)$ if defined, else $\rho^u(i)$ if defined, and otherwise be undefined.[5] The TF *corresponding to* $\mathcal{F}^u$ is $\langle \mathcal{X}, \mathcal{L}^-, \mathcal{L}^+, \mathcal{L}^0 \rangle$ such that:

$$\mathcal{L}^- = \{(i, a) \in \mathcal{L}^u | r^u(i) < 0\} \cup \{(a, i) \in \mathcal{L}^u | \mathcal{P}^u_{\mathcal{A}}(a) < 0\};$$
$$\mathcal{L}^+ = \{(i, a) \in \mathcal{L}^u | r^u(i) > 0\} \cup \{(a, i) \in \mathcal{L}^u | \mathcal{P}^u_{\mathcal{A}}(a) > 0\};$$
$$\mathcal{L}^0 = \{(i, a) \in \mathcal{L}^u | r^u(i) = 0\} \cup \{(a, i) \in \mathcal{L}^u | \mathcal{P}^u_{\mathcal{A}}(a) = 0\}.$$

Here, r^u and $\mathcal{P}^u_{\mathcal{A}}$ are used to determine the polarity of an affecting argument's effects on affected arguments.

For illustration, the TF corresponding to the directed A-I $\mathcal{F}^u$ for user u obtained from the A-I shown in Figure 1 is visualised in Figure 2. Here, there are no arguments affecting f_2 since it is rated by u. Given that this rating is negative and all aspects linked to f_2 are not rated by u, f_2 attacks all such aspects. Conversely, f_1 is not rated by u but has a positive rating from v and thus f_1 supports all (linked) aspects without a rating, i.e. a_1 and g_3. The fact that f_1 is not rated by u means that all aspects linked to f_1 affect it. Note that, in general, the neutralising relation is needed to represent the diluting effect of arguments on other arguments. For example, consider a movie f_1 with $n > 1$ linked aspects $\mathcal{L}(f_1) = \{a_1, \ldots, a_n\}$ such that $\mathcal{P}^u_{\mathcal{A}}(a_1) = 1$ and $\forall i > 1$ $\mathcal{P}^u_{\mathcal{A}}(a_i) = 0$, and movies f_2, f_3 with $\mathcal{L}(f_2) = \{a_1\}$, $\mathcal{L}(f_3) = \{a_2, \ldots, a_n\}$. The impact of the aspects on $\mathcal{P}^u_{\mathcal{A}}(f_2)$ should be greater than that on $\mathcal{P}^u_{\mathcal{A}}(f_1)$ (given that all of f_2's linked aspects have maximum predicted rating) - thus we need dialectical relations from $a_2, \ldots, a_n$ which reduce the strength of f_1. Moreover, the impact of the aspects on $\mathcal{P}^u_{\mathcal{A}}(f_3)$ should be null (given that all of f_3's aspects have neutral predicted rating) - thus the dialectical relations from $a_2, \ldots, a_n$ cannot be attacks. We use a neutralising relation that only dilutes the positive effect of a_1 so that the estimation of whether our user (dis)likes $a_2, \ldots, a_n$ does not decrease or increase our estimation of whether the user (dis)likes f_3 nor does it necessarily decrease our estimation of whether the user (dis)likes f_1.

If $\mathcal{P}^u_{\mathcal{X}}$ is taken to be a strength $\sigma : \mathcal{X} \mapsto [-1, 1]$ for arguments in $\mathcal{X}$, the TF corresponding to $\mathcal{F}^u$ is guaranteed to satisfy the following simple but intuitive property, which is a generalisation to the setting of TFs of one of the implications of 'strict balance' in [Baroni *et al.*, 2018].

Definition 10 A TF $\langle \mathcal{X}, \mathcal{L}^-, \mathcal{L}^+, \mathcal{L}^0 \rangle$ satisfies the *property of weak balance* if, for any $x, y \in \mathcal{X}$:
- if $\mathcal{L}^-(y) = \{x\}$, $\mathcal{L}^+(y) = \varnothing$ and $\mathcal{L}^0(y) = \varnothing$ then $\sigma(y) < 0$;
- if $\mathcal{L}^-(y) = \varnothing$, $\mathcal{L}^+(y) = \{x\}$ and $\mathcal{L}^0(y) = \varnothing$ then $\sigma(y) > 0$;
- if $\mathcal{L}^-(y) = \varnothing$, $\mathcal{L}^+(y) = \varnothing$ and $\mathcal{L}^0(y) = \{x\}$ then $\sigma(y) = 0$.

Proposition 1 Given the TF corresponding to any $\mathcal{F}^u$, $\sigma = \mathcal{P}^u_{\mathcal{X}}$ satisfies the property of weak balance.

The TF corresponding to $\mathcal{F}^u$ can be used to form explanations for predicted ratings, i.e. sub-graphs of the TF, that users can naturally interact with for providing feedback when presented with inaccurate predictions. In the remainder of

[5] It is easy to see, by definition of $\mathcal{L}^u$, that if $\exists (i, a) \in \mathcal{L}^u$, then $r^u(i)$ is defined.

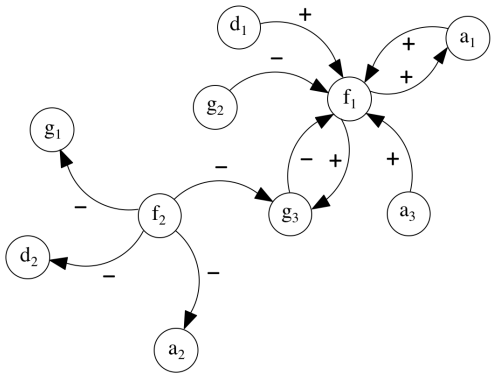

Figure 2: A graphical representation of the TF corresponding to the directed A-I $\mathcal{F}^u$ for user u from A-I in Figure 1. Here, '+' indicates 'support' ($\mathcal{L}^+$ in $\mathcal{F}^u$) and '-' indicates 'attack' ($\mathcal{L}^-$ in $\mathcal{F}^u$). Note that, in this simple illustration, there are no neutralisers ($\mathcal{L}^0$ in $\mathcal{F}^u$).

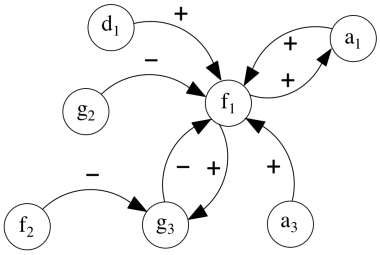

Figure 3: Example explanation for the recommendation of f_1 to u (based on the predicted rating of 0.21) using the TF in Figure 1.

this section we illustrate explanations and the feedback that they may help elicit from users.

As an illustration, in the context of our running example, let us assume that u is presented with a recommendation to watch f_1 (based on the computed, positive predicted rating of 0.21 for u), but the user is unhappy with the recommendation. Given that u presumably wants to reduce the chance of items like f_1 being recommended in the future, it would be desirable that the feedback from u would help reduce f_1's predicted rating (and thus items like f_1 since they would share aspects and similar users' ratings), if possible. Consider the graph in Figure 3 (which is a sub-graph of that in Figure 2): this can be seen as a qualitative explanation for the recommendation, indicating in particular which aspects affected the recommendation. A refined explanation may be as in Figure 4, generated (directly from the graph in Figure 2 as a sub-graph of the graph in Figure 3) if the aspects of type directors have the largest positive effect on f_1's predicted rating, i.e.[6]

$$\text{directors} = argmax_{t \in \mathcal{T}}(\mu_t[\sum_{a \in \mathcal{L}^u_t(f_1)} \mathcal{P}^u_{\mathcal{A}}(a)]/|\mathcal{L}^u_t(f_1)|) > 0$$

In the refined explanation, aspects of type directors are given prominence. User feedback on the (prominent) aspects

[6] $argmax_{s \in S} f(s) > 0 = \{s \in S | f(s) > 0 \wedge \forall t \in S \setminus \{s\} : f(t) \leq f(s)\}$.

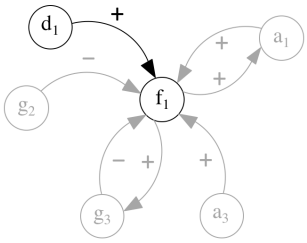

Figure 4: Another example explanation when the type directors has the largest positive effect on f_1's predicted rating.

affecting f_1 in Figure 3 or in Figure 4 may then turn into modifications of the constants used in the A-I method and result in improved future predictions. For example, if user u states that 'Directors of movies are not important to me', then the recommender system may opt for decreasing $\mu^u_{directors}$. In some cases this will guarantee improved predictions in the future, for example if the similarity constants do not change and the collaborative filtering had a minor effect relative to the aspects, namely

$$\mu^u_{c.f.}\rho^u(f_1) < \sum_{t \in \mathcal{T}} \mu_t[\sum_{a \in \mathcal{L}^u_t(f_1)} \mathcal{P}^u_{\mathcal{A}}(a)]/|\mathcal{L}^u_t(f_1)|$$

then decreasing $\mu^u_{directors}$ decreases $\mathcal{P}^u_{\mathcal{I}}(f_1)$.

Several other forms of explanation and feedback can be obtained from (sub-graphs of) argumentation graphs (omitted here for lack of space). Thus, these (sub-)graphs can be seen as providing a 'back-end' for a variety of explanations in different formats (e.g. graphical, visual or linguistic) for different contexts and types of users. For example, from Figure 4 one could draw a linguistic explanation 'f_1 is recommended for the most part due to its director'. If unhappy with the recommendation or the justification, the user could then react by reducing the effect of the director aspect, or by requesting the reasons that the system believes that this aspect should have such an impact: this may in turn require looking at a larger graph and provide feedback on the extended explanation. The feedback could be provided by selecting amongst predetermined responses with predetermined follow-up actions by the system (e.g. a modification to the user profile, as shown in the example, or to the ratings of aspects, in particular overriding predicted ratings with actual ratings). Note that, in some cases, the feedback may adversely affect accurate recommendations, for example if $\mu^u_{directors}$ is decreased the predicted rating of movies that the user would rate highly may decrease in turn. We posit that if the user explicitly states that this aspect is unimportant, then other aspects should be recalibrated to recover recommendations. Overall, our method could be integrated within an iterative process of explanation and feedback that could lead to overriding predicted with actual ratings of aspects that matter to a user, giving a recommender system which adapts to a user's feedback effectively.

6 Conclusions

We have proposed a hybrid method for making predictions in recommender systems, shown experimentally that it is com-

petitive in the movie domain, and illustrated how it can be used to generate effective explanations based on an argumentative reading of the framework via the method. Our method could be improved in several directions, for example, we started with arbitrary values for the parameters in our user profiles, but envisage improvements in accuracy by generating the constants systematically and optimally (e.g. by learning on bootstrapping), by allowing user-tailored tuning of the constants via feedback and by taking into account that users' tastes and preferences change over time. We also plan to conduct further experiments in the movie as well as other domains. From the argumentation view-point it would be useful to provide a dialectical re-interpretation of the notion of strength given by our method for predicting ratings, and identify additional properties of this notion of strength. Further, we plan to conduct in the future a systematic overview and user studies to ascertain suitability of the types of explanation and feedback that our method could support.

Acknowledgments

F. Toni was partially funded by the EPSRC project ROAD2H.

References

[Aggarwal, 2016] Charu C. Aggarwal. *Recommender Systems - The Textbook*. Springer, 2016.

[Altman, 1992] Naomi S. Altman. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3):175–185, 1992.

[Baroni *et al.*, 2017] Pietro Baroni, Giulia Comini, Antonio Rago, and Francesca Toni. Abstract Games of Argumentation Strategy and Game-Theoretical Argument Strength. In *PRIMA*, pages 403–419, 2017.

[Baroni *et al.*, 2018] Pietro Baroni, Antonio Rago, and Francesca Toni. How Many Properties Do We Need for Gradual Argumentation? In *AAAI*, 2018.

[Billsus and Pazzani, 1998] Daniel Billsus and Michael J. Pazzani. Learning Collaborative Information Filters. In *ICML*, pages 46–54, 1998.

[Briguez *et al.*, 2014] Cristian E. Briguez, Maximiliano C. Budán, Cristhian A. D. Deagustini, Ana G. Maguitman, Marcela Capobianco, and Guillermo R. Simari. Argument-based mixed recommenders and their application to movie suggestion. *Exp. Sys. with Applications*, 41(14):6467–6482, 2014.

[Burke, 2002] Robin D. Burke. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[Burke, 2003] Robin D. Burke. Hybrid Systems for Personalized Recommendations. In *Intelligent Techniques for Web Personalization, IJCAI Workshop*, pages 133–152, 2003.

[Cayrol and Lagasquie-Schiex, 2005] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *ECSQARU*, pages 378–389, 2005.

[Chesñevar *et al.*, 2009] Carlos I. Chesñevar, Ana G. Maguitman, and María P. González. Empowering Recommendation Technologies Through Argumentation. In *Argumentation in Artificial Intelligence*, pages 403–422. Springer, 2009.

[Czarkowski, 2006] Marek Czarkowski. *A scrutable adaptive hypertext*. PhD thesis, University of Sydney, Australia, 2006.

[Dung, 1995] Phan M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321 – 357, 1995.

[Gabbay, 2016] Dov M. Gabbay. Logical foundations for bipolar and tripolar argumentation networks: preliminary results. *J. Logic and Computation*, 26(1):247–292, 2016.

[García and Simari, 2004] Alejandro J. García and Guillermo R. Simari. Defeasible Logic Programming: An Argumentative Approach. *TPLP*, 4(1-2):95–138, 2004.

[George and Merugu, 2005] Thomas George and Srujana Merugu. A Scalable Collaborative Filtering Framework Based on Co-Clustering. In *ICDM*, pages 625–628, 2005.

[Hug, 2017] Nicolas Hug. Surprise, a Python library for recommender systems. http://surpriselib.com, 2017.

[Koren *et al.*, 2009] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*, 42(8):30–37, 2009.

[Lemire and Maclachlan, 2007] Daniel Lemire and Anna Maclachlan. Slope One Predictors for Online Rating-Based Collaborative Filtering. *CoRR*, abs/cs/0702144, 2007.

[Luo *et al.*, 2014] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems. *IEEE Trans. Ind. Inf.*, 10(2):1273–1284, 2014.

[Resnick and Varian, 1997] Paul Resnick and Hal R. Varian. Recommender Systems. *Comm. ACM*, 40(3):56–58, 1997.

[Teze *et al.*, 2015] Juan C. Teze, Sebastian Gottifredi, Alejandro J. García, and Guillermo R. Simari. Improving argumentation-based recommender systems through context-adaptable selection criteria. *Exp. Sys. with Applications*, 42(21):8243–8258, 2015.

[Tintarev and Masthoff, 2007] Nava Tintarev and Judith Masthoff. A Survey of Explanations in Recommender Systems. In *ICDE Workshops*, pages 801–810, 2007.

[Töscher *et al.*, 2009] Andreas Töscher, Michael Jahrer, and Robert M. Bell. The BigChaos Solution to the Netflix Grand Prize, 2009.

[Vozalis and Margaritis, 2006] Manolis G. Vozalis and Konstantinos G. Margaritis. Applying SVD on Generalized Item-based Filtering. *Int. J. on Comp. Sc. and Appl.*, 3(3):27–51, 2006.